Document-Level Event Argument Extraction by Conditional Generation

Sha Li and **Heng Ji** and **Jiawei Han**
University of Illinois at Urbana-Champaign, IL, USA
{shal2, hengji, hanj}@illinois.edu

Abstract

Event extraction has long been treated as a sentence-level task in the IE community. We argue that this setting does not match human information seeking behavior and leads to incomplete and uninformative extraction results. We propose a document-level neural event argument extraction model by formulating the task as conditional generation following event templates. We also compile a new document-level event extraction benchmark dataset WIKIEVENTS which includes complete event and coreference annotation. On the task of argument extraction, we achieve an absolute gain of 7.6% F1 and 5.7% F1 over the next best model on the RAMS and WIKIEVENTS datasets respectively. On the more challenging task of informative argument extraction, which requires implicit coreference reasoning, we achieve a 9.3% F1 gain over the best baseline. To demonstrate the portability of our model, we also create the first end-to-end zero-shot event extraction framework and achieve 97% of fully supervised model's trigger extraction performance and 82% of the argument extraction performance given only access to 10 out of the 33 types on ACE. [1]

1 Introduction

By converting a large amount of unstructured text into trigger-argument structures, event extraction models provide unique value in assisting us process volumes of documents to form insights. While real-world events are often described throughout a news document (or even span multiple documents), the scope of operation for existing event extraction models have long been limited to the sentence level.

Early work on event extraction originally posed the task as document level role filling (Grishman and Sundheim, 1996) on a set of narrow scenarios and evaluated on small datasets. The release of ACE[2], a large scale dataset with complete event annotation, opened the possibility of applying powerful machine learning models which led to substantial improvement in event extraction. The success of such models and the widespread adoption of ACE as the training dataset established sentence-level event extraction as the mainstream task defintion.

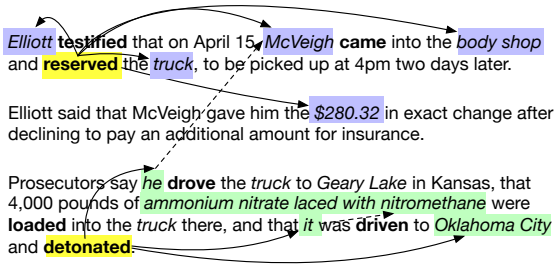

Figure 1: Two examples of cross-sentence inference for argument extraction from our WIKIEVENTS dataset. The PaymentBarter argument of the Transaction.ExchangeBuySell event triggered by "reserved" in the first sentence can only be found in the next sentence. The Attack.ExplodeDetonate event triggered by "detonated" in the third sentence has an uninformative argument "he", which needs to be resolved to the name mention "McVeigh" in the previous sentences.

This formulation signifies a misalignment between the information seeking behavior in real life and the exhaustive annotation process in creating the datasets. An information seeking session (Mai, 2016) can be divided into 6 stages: task initiation, topic selection, pre-focus exploration, focus information, information collection and search closure (Kuhlthau, 1991). Given a target event ontology, we can safely assume that topic selection is complete and users start from skimming the documents before they discover events of interest, focus on such events and then aggregate all relevant information for the events. In both the "pre-focus

[1]The programs, data and resources are publicly available for research purpose at https://github.com/raspberryice/gen-arg.

[2]https://www.ldc.upenn.edu/collaborations/past-projects/ace

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 894–908
June 6–11, 2021.

exploration" and "information collection" stages, users naturally cross sentence boundaries.

Empirically, using sentence boundaries as event scopes conveniently simplifies the problem, but also introduces fundamental flaws: the resulting extractions are *incomplete* and *uninformative*. We show two examples of this phenomenon in Figure 1. The first example exemplifies the case of implicit arguments across sentences. The sentence that contains the PaymentBarter argument "$280.32" is not the sentence that contains the trigger "reserve" for the ExchangeBuySell event. Without a document-level model, such arguments would be missed and result in *incomplete* extraction. In the second example, the arguments are present in the same sentence, but written as pronouns. Such extraction would be *uninformative* to the reader without cross-sentence coreference resolution.

We propose a new end-to-end document-level event argument extraction model by framing the problem as *conditional generation* given a template. Conditioned on the unfilled template and a given context, the model is asked to generate a filled-in template with arguments as shown in Figure 2. Our model does not require entity recognition nor coreference resolution as a preprocessing step and can work with long contexts beyond single sentences. Since templates are usually provided as part of the event ontology definition, this requires no additional human effort. Compared to recent efforts (Du and Cardie, 2020; Feng et al., 2020; Chen et al., 2020) that retarget question answering (QA) models for event extraction, our generation-based model can easily handle the case of missing arguments and multiple arguments in the same role without the need of tuning thresholds and can extract all arguments in a single pass.

In order to evaluate the performance of document-level event extraction, we collect and annotate a new benchmark dataset WIKIEVENTS. This document-level evaluation also allows us to move beyond the nearest mention of the argument and instead seek the most informative mention[3] in the entire document context. In particular, only 34.5% of the arguments detected in the same sentence as the trigger can be considered informative. We present this new task of *document-level informative argument extraction* and show that while this task requires much more cross-sentence inference, our model can still perform reliably well.

Since we provide the ontology information (which roles are needed for the event) through the template as an external condition, our model has excellent portability to unseen event types. By pairing up our argument extraction model with a keyword-based zero-shot trigger extraction model, we enable zero-shot transfer for new event types.

The major contributions of this paper can be summarized as follows:

1. We address the document-level argument extraction task with an end-to-end neural event argument extraction model by conditional text generation. Our model does not rely on entity extraction nor entity/event coreference resolution. Compared to QA-based approaches, it can easily handle missing arguments and multiple arguments in the same role.

2. We present the first *document-level event extraction* benchmark dataset with *complete event and coreference annotation*. We also introduce the new *document-level informative argument extraction* task, which evaluates the ability of models to learn entity-event relations over long ranges.

3. We release the first end-to-end zero-shot event extraction framework by combining our argument extraction model with a zero-shot event trigger classification model.

2 Method

The event extraction task consists of two subtasks: trigger extraction and argument extraction. The set of possible event types and roles for each event type are given by the event ontology as part of the dataset. One template for each event type is usually pre-defined in the ontology. [4]

We first introduce our document-level argument extraction model in Section 2.1 and then introduce our zero-shot keyword-based trigger extraction model in Section 2.2.

2.1 Argument Extraction Model

We use a conditional generation model for argument extraction, where the condition is an unfilled template and a context. The template is a sentence

[3] We prefer name mentions over nominal mentions and only use pronoun mentions when no other mentions exist.

[4] ACE does not come with templates, but since the event types are subsets of the RAMS AIDA ontology and the KAIROS ontology, we reused templates from the these ontologies.

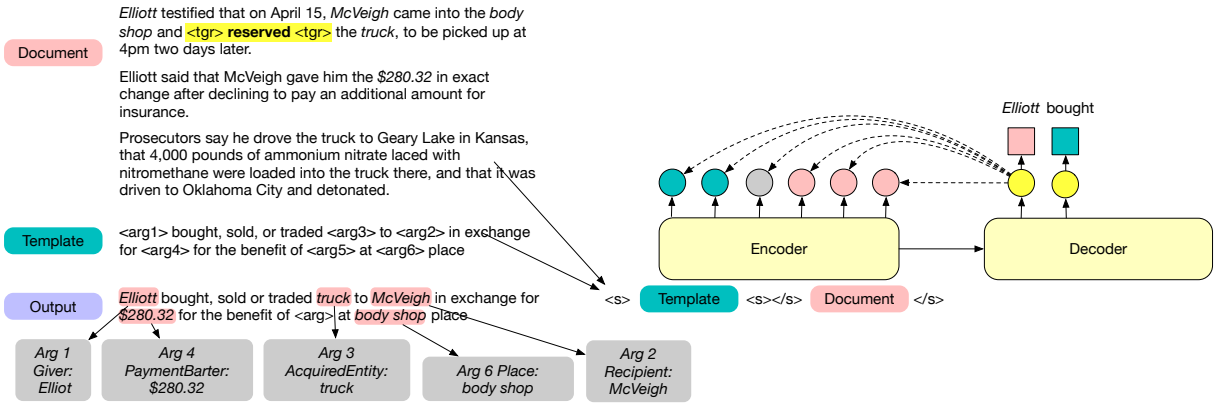

Figure 2: Our argument extraction model using conditional generation. On the left we show an example document, template and the desired output for the instance. Each example document may contain multiple event triggers and we use special $\langle tgr \rangle$ tokens to markup the target event trigger for argument extraction (the highlighted word "reserved"). The input to the model is the concatenation of the template and the document. The decoded tokens are either from the template or the document. The color of the generated tokens indicate its copy source. After the filled template is generated, we extract the spans to produce the final output.

that describes the event with $\langle arg \rangle$ placeholders. The generated output is a filled template where placeholders are replaced by concrete arguments. An example of the unfilled template from the ontology and the filled template for the event type Transaction.ExchangeBuySell [5] can be seen in Figure 2. Notably, one template per event type is given in the ontology, and does not require further human curation as opposed to the question designing process in question answering (QA) models (Du and Cardie, 2020; Feng et al., 2020).

Our base model is an encoder-decoder language model (BART (Lewis et al., 2020), T5 (Raffel et al., 2020). The generation process models the conditional probability of selecting a new token given the previous tokens and the input to the encoder.

$$p(x \mid c) = \prod_{i=1}^{|x|} p\left(x_i \mid x_{<i}, c\right) \quad (1)$$

In the encoder, bidirectional attention layers are used to enable interaction between every pair of tokens and produce the encoding for the context c. Each layer of the decoder performs cross-attention over the output of the encoder in addition to the attention over the previous decoded tokens.

To utilize the encoder-decoder LM for argument extraction, we construct an input sequence of $\langle s \rangle$ template $\langle s \rangle \langle /s \rangle$document $\langle /s \rangle$. All argument names (arg1, arg2, etc.) in the template are replaced by a special placeholder token $\langle arg \rangle$. The ground truth sequence is the filled template where the placeholder token is replaced by the argument span whenever possible. In the case where there are multiple arguments for the same slot, we connect the arguments with the word "and".

The generation probability is computed by taking the dot product between the decoder output and the embeddings of tokens from the input.

$$\mathrm{p}(\mathrm{x}_i = w | x_{<i}, c, t) = \begin{cases} \text{Softmax}\left(h_i^T \text{Emb}(w)\right) & w \in V_c \\ 0 & w \notin V_c \end{cases} \quad (2)$$

To prevent the model from hallucinating arguments, we restrict the vocabulary of words to V_c: the set of tokens in the input.

The model is trained by minimizing the negative loglikelihood over all (content, template, output) instances in the dataset D:

$$\mathcal{L}(D) = -\sum_{i=1}^{|D|} \log p_\theta \left(x^i \mid c^i\right) \quad (3)$$

The event ontology often imposes entity type constraints on the arguments. When using the template only, the model has no access to such constraints and can generate seemingly fluent and sensible responses with the wrong arguments. Inspired by (Shwartz et al., 2020), we use clarification statements to add back constraints without breaking the end-to-end property of the model.

In the example presented in Table 1, we can see that the greedy decoding selects "tax plan" as the second Participant argument for the PublicStatement event. Apart from the preposition "with",

[5] This type is used for a transaction transferring or obtaining money, ownership, possession, or control of something, applicable to any type, nature, or method of acquisition including barter.

Context	Original	After
When outlining her tax reform policy , Clinton has made clear that she wants to tax the wealthy and make sure they "pay their fair share ." **She** has proposed (PublicStatement) a **tax plan** that would require millionaires and billionaires to pay more taxes than middle-class and lower -income individuals.	**She** communicated with **tax plan** about ⟨ arg ⟩ at ⟨ arg ⟩ place. **She** is a person/organization/country. **tax plan** is a person/organization/country.	**She** communicated with ⟨ arg ⟩ about **tax plan** at ⟨ arg ⟩ place. **She** is a person/organization/country.

Table 1: Example of adding type constraints through clarification. The Participant argument of PublicStatement event can only be a person, organization or geo-political entity. The Topic argument can be any type of entity.

there is nothing in the template indicating that this slot should be filled in with a person instead of a topic. To remedy this mistake, we append "clarifications" for its argument fillers in the form of type statements: ⟨ arg ⟩ is a ⟨ type ⟩. We then rerank the candidate outputs by the language modeling probability of the filled template and clarifications. When there are multiple valid types, we take the maximum probability of the valid type statements.

$$\log p(x|c) = \sum_i \log p(x_i|x_{<i}, c) + \max_{e \in E_r} \log p(z_e|x, c) \quad (4)$$

E_r is the set of valid entity types for the role r according to the ontology and z_e is the type statement. Since "tax plan is a person." goes against commonsense, the probability of generating this sentence will be low. In this way, we can prune responses with conflicting entity types.

2.2 Keyword-Based Trigger Extraction Model

Our argument extraction model relies on detected event triggers (type and offset) as input. Any trigger extraction model could be used in practice, but here we describe a trigger extraction model designed to work with only keyword-level supervision. For example for the "StartPosition" event, we use 3 keywords "hire, employ and appoint" as initial supervision with no mention level annotation. [6] This module allows quick transfer to new event types of interest.

We treat the trigger extraction task as sequence labeling and our model is an adaptation of TapNet (Yoon et al., 2019; Hou et al., 2020), which was designed for few-shot classification and later extended to Conditional Random Field (CRF) models. Compared with (Hou et al., 2020), we do not collapse the entries of the transition matrix, making it possible for our model to learn different probabilities for each event type. Since our model takes class keywords as input, we refer to this model as TapKey.

For each event type, we first obtain a class representation vector c_k based on given keywords using the masked category prediction method in (Meng et al., 2020). This class representation vector is an average over the BERT vector representations of the keywords, with some filtering applied to remove ambiguous occurrences. Details of the filtering process are included in Appendix A.

Following the linear-chain CRF model, the probability of a tagged sequence is:

$$\log p(y|h;\theta) \propto \sum_i \varphi(y_i|h_i) + \sum_i \psi(y_i|y_{i-1}, h_i) \quad (5)$$

h_i is the output of the embedding network (in our case, BERT-large) corresponding to x_i.

The label space for y_i is the set of IO tags. We choose to use this simplified tagging scheme because it has fewer parameters and the fact that consecutive triggers of the same event are very rare.

The feature function $\varphi(\cdot)$ is defined as

$$\varphi(y_i = k|h_i) = \text{Softmax}\left(M(h_i)^T M(\phi_k)\right) \quad (6)$$

ϕ_k is a normalized reference vector for class k and M is a projection matrix, both of which are parameters of the model. M is not a learned parameter, but solved by taking the QR decomposition of a modified reference vector matrix. Specifically, M satisfies the following equation:

$$M^T(c_k - \lambda\hat{\phi}_k) = 0 \quad (7)$$

We refer to the TapNet (Yoon et al., 2019) paper for details and also provide a simplified derivation in Appendix A.

The transition score $\psi(\cdot)$ between tags is parameterized using two diagonal matrices W and W_o:

$$\psi(y_i = k|y_{i-1} = l, h_i) = \begin{cases} M(\phi_k)WM(h_i) & k = l \neq 0 \\ M(\phi_k)W_oM(h_i) & k \text{ or } l = 0 \\ 0 & k \neq l \end{cases} \quad (8)$$

[6] In a fully supervised setting, it might be desirable to use a supervised trigger extraction model for optimal performance.

	Train	Dev	Test
# Event types	49	35	34
# Arg types	57	32	44
# Docs	206	20	20
# Sentences	5262	378	492
# Events	3241	345	365

Table 2: Statistics for the WIKIEVENTS dataset.

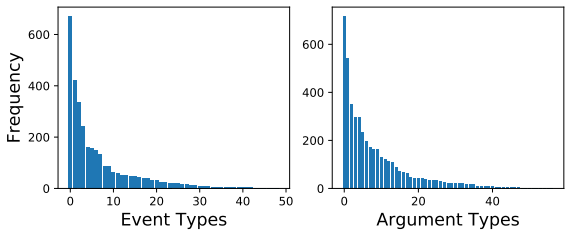

Figure 3: Distribution of event types and argument types in the WIKIEVENTS dataset.

In the training stage, the model parameters $\{\phi, W, \theta_f\}$ are learned by minimizing the negative log probability of the sequences.

$$\mathcal{L} = -\frac{1}{N}\sum \log p(y|h;\theta) + \alpha\|\Phi^T\Phi - I\|^2 \quad (9)$$

The matrix Φ of all reference vectors is initialized as a diagonal matrix and the second term regularizes the vectors to be close to orthonormal during training. α is a hyperparameter.

In the zero-shot setting, we first train on pseudo labeled data before we apply the model. In the pseudo labeling stage, we directly use the cosine similarity between class vectors and the embeddings of the tokens from the language model to assign labels to text. We only use labels with high confident for both event I tags and O tags. The remainder of the tokens will be tagged as X for unknown. Then we train the model on the token classification task. Since none of the parameters in the model are class-specific, the model can be used in a zero-shot transfer setting.

3 Benchmark Dataset WIKIEVENTS

3.1 Evaluation Tasks

Our dataset evaluates two tasks: *argument extraction* and *informative argument extraction*.

For *argument extraction*, we use head word F1 (Head F1) and coreferential mention F1 (Coref F1) as metrics. We consider an argument span to be correctly identified if the offsets match the reference. If the argument role also matches, we consider the argument is correctly classified. Since annotators

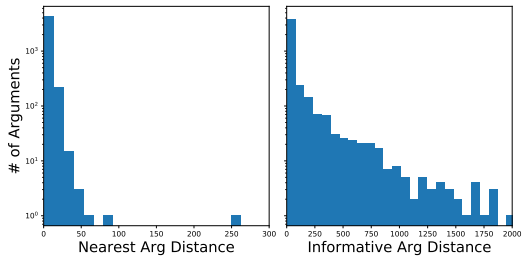

Figure 4: Distribution of distance between event trigger and arguments. Distance is measured in number of words.

are asked to annotate the head word when possible, we refer to this metric as Head F1. For Coref F1, the model is given full credit if the extracted argument is coreferential with the gold-standard argument as used in (Ji and Grishman, 2008).

For downstream applications such as knowledge base construction and question answering, argument fillers that are pronouns will not be useful to the user. Running an additional coreference resolution model to resolve them will inevitably introduce propagation errors. Hence, we propose a new task: *document-level informative argument extraction*. We define name mentions to be more informative than nominal mentions, and pronouns to be the least informative. When the mention type is the same, we select the longest mention as the most informative one. Under this task, the model will only be given credit if the extracted argument is the most informative mention in the entire document.

3.2 Dataset Creation

We collect English Wikipedia articles that describe real world events and then follow the reference links to crawl related news articles. We first manually identify category pages such as https://en.wikipedia.org/wiki/Category:Improvised_explosive_device_bombings_in_the_United_States and then for each event page (i.e. https://en.wikipedia.org/wiki/Boston_Marathon_bombing), we record all the links in its "Reference" section and use an article scraping tool[7] to extract the full text of the webpage.

We follow the recently established ontology from the KAIROS project[8] for event annotation. This ontology defines 67 event types in a three level hierarchy. In comparison, the commonly used

[7] https://github.com/codelucas/newspaper

[8] https://www.ldc.upenn.edu/collaborations/current-projects

ACE ontology has 33 event types defined in two levels.

We hired graduate students as annotators and provided example sentences for uncommon event types. A total of 26 annotators were involved in the process. We used the BRAT[9] interface for online annotation.

The annotation process is divided into 2 stages: event mention (trigger and argument) annotation and event coreference annotation. In addition to coreferential mention clusters, we also provide the most informative mention for each cluster. Details about the data collection and annotation process can be found in Appendix B.

3.3 Dataset Analysis

Overall statistics of the dataset are listed in Table 2. Compared to ACE, our WIKIEVENTS dataset has a much richer event ontology, especially for argument roles. The observed distributions of event types and argument roles are shown in Figure 3.

We further examine the distance between the event trigger and arguments in Figure 4. When considering the nearest argument mention, the distribution of the arguments is very concentrated towards 0, showing that this annotation standard favors local extractions. In the case of extracting informative mentions, we have a relatively flat long tail distribution with the average distance being 68.82 words (compared to 4.75 words for the nearest mention). In particular, only 34.5% of the arguments detected in the same sentence as the trigger can be considered informative. This confirms the need for document level inference in the search of informative argument fillers.

4 Experiments

Our experiments fall under three settings: (1) document-level event argument extraction; (2) document-level informative argument extraction and (3) zero-shot event extraction.

For document-level event argument extraction we follow the conventional approach of regarding the argument mention with closest proximity to the trigger as the ground truth. In the second setting we consider the most informative mention of the argument as the ground truth.

The zero-shot setting examines the portability of the model to new event types. Under this setting we consider a portion of the event types to be known and only annotation for these event types will be seen. We used two settings for selecting known types: 10 most frequent events types and 8 event types, one from each parent type of the event ontology. The evaluation is done on the complete set of event types. We refer the reader to Appendix C for implementation details and hyperparameter settings.

Split	Event Types	# Sents	# Events
Full Training	33	17172	4202
Freq	10	17172	3398
Ontology	8	17172	1311
Dev	-	923	450
Test	-	832	403

Table 3: Dataset statistics for ACE under multiple settings. In the **Freq** split, we keep the 10 most frequent event types. In the **Ontology** split, we keep 1 event subtype per general type in LIFE, MOVEMENT, TRANSACTION, BUSINESS, CONFLICT, CONTACT, PERSONNEL and JUSTICE. [13]

4.1 Datasets

In addition to our dataset WIKIEVENTS, we also report the performance on the Automatic Content Extraction (ACE) 2005 dataset[10] and the Roles Across Multiple Sentences (RAMS) dataset[11].

We follow preprocessing from (Lin et al., 2020; Wadden et al., 2019) for the ACE dataset. [12] Statistics of the ACE data splits can be found in Table 3. RAMS (Ebner et al., 2020) is a recently released dataset with cross-sentence argument annotation. A 5-sentence window is provided for each event trigger and the closest argument span is annotated for each role. We follow the official data splits from Version 1.0.

4.2 Document-Level Event Argument Extraction

Table 4 shows the performance for argument extraction on RAMS. On the RAMS dataset, we mainly compare with Two-step (Zhang et al., 2020), which is the current SOTA on this dataset. To handle long contexts, it breaks down the argument extraction into two steps: head detection and expansion.

In Table 5 we show the results for the WIKIEVENTS dataset. We compare with a pop-

[9] https://brat.nlplab.org/

[10] https://www.ldc.upenn.edu/collaborations/past-projects/ace

[11] http://nlp.jhu.edu/rams

[12] Note that our preprocessing procedure is slightly different from (Du and Cardie, 2020) as we kept pronouns as valid event triggers and arguments.

Model	Span F1	Head F1
Seq	40.5	48.0
Two-step	41.8	49.7
BART-Gen	**48.64**	**57.32**

Table 4: Supervised argument extraction results (%) on RAMS test set. Both the **Seq** and **Two-step** model use type-constrained decoding to improve performance. The official scorer was used to compute results.

Model	Arg Identification		Arg Classification	
	Head F1	Coref F1	Head F1	Coref F1
BERT-CRF	69.83	72.24	54.48	56.72
BERT-QA	61.05	64.59	56.16	59.36
BERT-QA-Doc	39.15	51.25	34.77	45.96
BART-Gen	**71.75**	**72.29**	**64.57**	**65.11**

Table 5: Argument extraction results (%) on WIKIEVENTS test set.

ularly used BERT-CRF baseline (Shi and Lin, 2019) that performs trigger extraction on sentence-level and BERT-QA (Du and Cardie, 2020) ran on sentence-level and document-level.

4.3 Document-Level Informative Argument Extraction

We test on WIKIEVENTS using the informative argument as the training data and also compare with the BERT-CRF and BERT-QA baselines. Results are shown in Table 6.

Comparing the results in Tables 5 and 6, we have the following findings:

1. Informative argument extraction is a much more difficult task compared to nearest argument extraction. This is exemplified by the large performance gap for all models.

2. While CRF models are good at identifying spans, the performance is hindered by classification. The arguments follow a long tail distribution and since CRF models learn each argument tag separately, it cannot leverage the similarity between argument roles to improve the performance on rarely seen roles.

3. QA models, on the other hand, suffer from poor argument identification. When the QA model produces multiple answers for the same role, these answer spans are often close to each other or overlap. We show a concrete example in the qualitative analysis.

4. Directly applying the BERT-QA model to document level does not work. The QA model gets easily distracted by the additional context and does not know which event to focus on. We think that this is not a fundamental limitation of the QA approach, but a sign that repurposing QA models for document-level event extraction needs more investigation.

Model	Arg-I		Arg-C	
	Head F1	Coref F1	Head F1	Coref F1
BERT-CRF	52.71	58.12	43.29	47.70
BERT-QA	46.88	49.89	43.44	46.45
BERT-QA-Doc	26.29	31.54	24.46	29.26
BART-Gen	**56.10**	**62.48**	**51.03**	**57.04**

Table 6: Informative argument extraction results on WIKIEVENTS test set.

4.4 Zero-Shot Event Extraction

We show results for the zero-shot transfer setting in Table 8. Since the baseline BERT-CRF model (Shi and Lin, 2019) cannot handle new labels directly, we exclude it from comparison. In addition to BERT-QA, we also replace our TAPKEY trigger extraction model with a Prototype Network(Snell et al., 2017)[14]. We replace the prototypes with the class vectors to enable zero-shot learning. Complete results for trigger extraction are included in Appendix D.

The performance of BERT-QA is greatly limited by the trigger identification step. Both the Prototype network and our model TAPKEY can leverage the keyword information to assist transfer. Remarkably, TAPKEY has only 3 points drop in F1 using only 30% of the training data compared to the full set. The argument extraction component is more sensitive to the reduction in training data, but still performs relatively well. We notice that when a template is completely new, the model might alter the template structure during generation.

4.5 Qualitative Analysis

We show a comparison of our model's extractions with baselines in Table 7 for the argument extraction task on WIKIEVENTS. Our model is able to effectively capture all arguments, while the CRF model struggles with rare event types and the QA model is hindered by over-generation.

An example of informative argument extraction from our model is displayed in Figure 6. Our model

[14]When 0 event types are seen, we set the transformation function in the Prototype Network to be an identity function; in other cases, we use a two layer MLP.

Context	Role	Ours	BERT-CRF	BERT-QA
I have been in touch (E1:Contact.Contact.Correspondence) with the NDS **official** in the province and they told me that over 100 **members** of the NDS were killed(E2:Life.Die) in the big explosion , " the former provincial official said . Sharif Hotak , a member of the provincial council in Maidan Wardak said **he** saw (E3:Cognitive.IdentifyCategorize) **bodies** of 35 Afghan forces in the **hospital**."	E1: Participant	I NDS official	I official	NDS I official NDS official
	E2: Victim	members	members	members
	E3: Identifier	he	N/A	N/A
	E3: IdentifiedObject	bodies	N/A	N/A
	E3: Place	hospital	N/A	N/A

Table 7: An example of document-level argument extraction task on WIKIEVENTS. This excerpt contains 3 events: Contact, Die and IdentifyCategorize. For the Contact event E1, BERT-QA over-generates answers for the participant span. For the Die event, all three models can correctly extract the Victim argument. For the IdentifyCategorize event which is relatively rare, only our model can successfully extract all arguments.

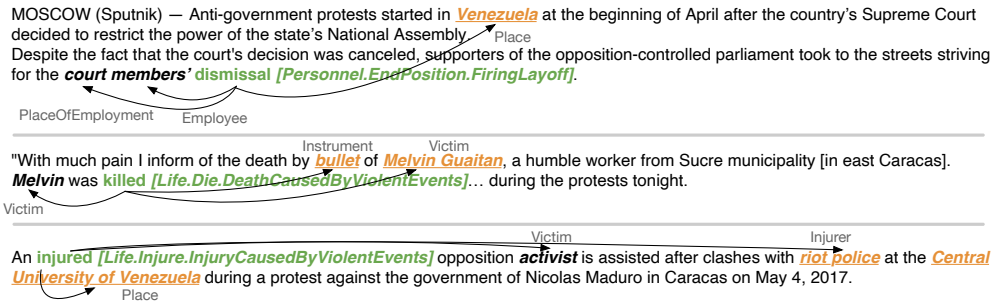

Figure 5: An example of our model's argument extraction output on the SM-KBP dataset. Arguments highlighted in orange are newly found by our model compared to the baseline system OneIE (Lin et al., 2020).

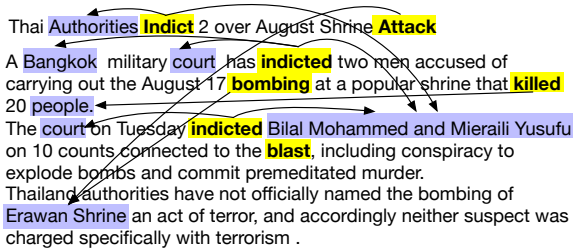

Figure 6: An example of our model's prediction of informative arguments. Only arguments for the bold-faced event triggers are shown. Notably, our model can correctly identify "Bilal Mohammed and Mieralli Yusufu" as the as the Defendant for all the ChargeIndict events and "Erawan Shrine" as the place of attack.

is able to choose the informative mentions of the Defendant of indiction and Place of attack even when the trigger is a few sentences away.

We also applied our model as part of a pipeline multimedia multilingual knowledge extraction system (Li et al., 2020) for the NIST streaming multimedia knowledge base population task (SM-KBP2020)[15]. Our model was able to discover 53% new arguments compared to the original system, especially for those that were further away from the event trigger. The overall system achieved top 1 performance. We show some examples in Figure 5.

4.6 Remaining Challenges

Ontological Constraints Some of the roles are mutually exclusive, such as the Origin/Destination in the Transport event and the Recipient/Yielder in the Surrender event. In the following example, "Japan" was extracted as both part of the Recipient and the Yielder of the Surrender event: *"If South Korea drifts into the orbit of the **US and Japan**, China's influence on the Korean peninsula could be badly compromised." At a military parade in Beijing to mark the 70th anniversary of the surrender of **Japan** last September, ..."*. Such constraints might be incorporated into the decoding process of the model.

Commonsense Knowledge In the following instance with implicit arguments: *"Whether the **U.S.** extradites **Gulen** or not this will be a political decision, "Bozdag said." If he is not extradited, **Turkey** will have been sacrificed for a terrorist." A recent opinion poll showed two thirds of Turks agree with their president that Gulen was behind the **coup plot.**"*, our model mistakenly labels "U.S." as the Destination of the extradition and "Turkey" as the

[15] https://tac.nist.gov/2020/KBP/SM-KBP/

# Seen Event Types	Model	TI F1	TC F1	AI Head F1	AC Head F1
0	Prototype	3.19	2.22	-	-
	TAPKEY	55.19	52.10	-	-
10 most frequent	BERT-QA	57.06	53.83	39.37	38.27
	Prototype + BART-Gen	69.48	66.06	46.08	41.93
	TAPKEY + BART-Gen	72.31	69.23	48.18	44.19
1 per general type	BERT-QA	27.56	25.32	24.87	24.29
	Prototype + BART-Gen	68.29	65.85	39.51	34.63
	TAPKEY + BART-Gen	72.23	68.55	39.80	35.11
Full	TAPKEY + BART-Gen	74.36	71.13	55.22	53.71

Table 8: Zero-shot event extraction results (%) on ACE. "10 most frequent event types" corresponds to the **Freq** data split and "1 per general type" corresponds to the **Ontology** data split. Fully supervised results are provided for reference.

Source even though the Extraditer is correctly identified as "U.S.". Commonsense knowledge such as "The extraditer, if being a country, is usually is same as the source of extradition" would be helpful to fix this error.

5 Related Work

5.1 Document-Level Event Extraction

Document-level event extraction can be traced back role filling tasks from the MUC conferences (Grishman and Sundheim, 1996) that required retrieving participating entities and attribute values for specific scenarios. The KBP slot filling challenge[16] is akin to this task, but centered upon entities.

In general, document-level argument extraction is an under-explored topic, mainly due to the lack of datasets. There have been a few datasets published specifically for implicit semantic role labeling, such as the SemEval 2010 Task 10 (Ruppenhofer et al., 2010), the Beyond NomBank dataset (Gerber and Chai, 2010) and ON5V (Moor et al., 2013). However, these datasets were small in size and only covered a small set of carefully selected predicates. Recently, (Ebner et al., 2020) published the RAMS dataset, which contains annotation for cross-sentence implicit arguments covering a wide range of event types. Albeit, this dataset only annotates one event per document, motivating us to create a new benchmark with complete event and coreference annotation.

The GRIT model (Du et al., 2021) is a generative model designed for the MUC task which can be seen as filling in predefined tables. In comparison, we treat the template (for example "<arg1> attacked <arg2> using <arg3> at <arg4> place") as part of the model input along with the document context. This allows us to share model parameters across all event types and enables zero-shot transfer to new event types.

5.2 Zero-shot Event Extraction

Early attempts at zero-shot or few-shot event extraction rely on preprocessing such as semantic role labeling(SRL) (Peng et al., 2016) or abstract meaning representation (AMR) (Huang et al., 2018) to detect trigger mentions and argument mentions before performing classification on the detected spans.

Another line of work only examines the subtask of trigger detection, essentially reducing the task to few-shot classification. Both (Lai et al., 2020) and (Deng et al., 2020) extend upon the prototype network model (Snell et al., 2017) for classification.

Recent work on zero-shot event extraction has posed the problem as question answering (Chen et al., 2020; Du and Cardie, 2020; Feng et al., 2020) with different ways of designing the questions.

6 Conclusion & Future Work

In this paper, we advocate document-level event extraction and propose the first document-level neural event argument extraction model. We also release the first document-level event extraction benchmark dataset WIKIEVENTS with complete event and coreference annotation. On both the conventional argument extraction task and the new informative argument extraction task, our proposed model surpasses CRF-based and QA-based baselines by a wide margin. Additionally, we demonstrate the portability of our model by applying it to the zero-shot setting. Going forward, we would like to incorporate more ontological knowledge to produce more accurate extractions.

[16] https://tac.nist.gov/2017/KBP/index.html

Acknowledgement

The authors would like to thank Tuan Lai and Zhenhailong Wang for their help in the data processing and annotation organization effort. We also thank all the annotators who have contributed to the annotations of our training data (in alphabetical order): Daniel Campos, Anthony Cuff, Yi R. Fung, Xiaodan Hu, Emma Bonnette Hamel, Samual Kriman, Meha Goyal Kumar, Manling Li, Tuan M. Lai, Ying Lin, Sarah Moeller, Ashley Nobi, Xiaoman Pan, Nikolaus Parulian, Adams Pollins, Rachel Rosset, Haoyu Wang, Qingyun Wang, Zhenhailong Wang, Spencer Whitehead, Lucia Yao, Pengfei Yu, Qi Zeng, Haoran Zhang, Hongming Zhang, Zixuan Zhang.

This research is based upon work supported by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and Air Force No. FA8650-17-C-7715. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, Online. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. *WSDM*.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Xinya Du, Alexander M. Rush, and Claire Cardie. 2021. Grit: Generative role-filler transformers for document-level event entity extraction.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction. In *ArXiv*.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Carol C Kuhlthau. 1991. Inside the search process: Information seeking from the user's perspective. *Journal of the American society for information science*, 42(5):361–371.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Manling Li, Ying Lin, Tuan Manh Lai, Xiaoman Pan, Haoyang Wen, Sha Li, Zhenhailong Wang, Pengfei Yu, Lifu Huang, Di Lu, Qingyun Wang, Haoran Zhang, Qi Zeng, Chi Han, Zixuan Zhang, Yujia Qin, Xiaodan Hu, Nikolaus Parulian, Daniel Campos, Heng Ji, Brian Chen, Xudong Lin, Alireza Zareian, Amith Ananthram, Emily Allaway, Shih-Fu Chang, Kathleen McKeown, Yixiang Yao, Yifan Wang, Michael Spector, Mitchell DeHaven, Daniel Napierski, Marjorie Freedman, Pedro Szekely, Haidong Zhu, Ram Nevatia, Yang Bai, Yifan Wang, Ali Sadeghian, Haodi Ma, and Daisy Zhe Wang. 2020. Gaia at sm-kbp 2020 - a dockerlized multi-media multi-lingual knowledge extraction, clustering, temporal tracking and hypothesis generation system. In *Proc. Text Analysis Conference (TAC2020)*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Jens-Erik Mai. 2016. *Looking for information: A survey of research on information seeking, needs, and behavior*. Emerald Group Publishing.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017, Online. Association for Computational Linguistics.

Tatjana Moor, Michael Roth, and Anette Frank. 2013. Predicate-specific annotations for implicit role binding: Corpus annotation, data analysis and evaluation experiments. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 369–375, Potsdam, Germany. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2010. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50, Uppsala, Sweden. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *NeurIPS*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Document-Level Event Argument Extraction by Conditional Generation: Appendix

Sha Li and **Heng Ji** and **Jiawei Han**
University of Illinois at Urbana-Champaign, IL, USA
{shal2, hengji, hanj}@illinois.edu

1 Trigger Extraction Model Details

1.1 Tagging Scheme

We use the IO tagging scheme, where I stands for "inside a span" and O stands for "outside any span". This simplified tagging scheme was selected to reduce parameters without much loss of modeling power since (1) triggers are often single words and I tags in the BIO (B stands for "beginning of a span") scheme are infrequent and (2) we rarely see two consecutive event triggers of the same type.

1.2 Class Vectors

For each of the event types, we provided 3 keywords as initial seeds. If the event type can be triggered by nominals, we additionally add keywords for the nominal form. Our chosen keywords will be provided along with the ontology file as supplementary materials.

For each event type, we search for its corresponding keywords' occurrence in the Gigaword corpus. To filter out ambiguous usages of the keywords, we apply BERT-large as a masked language model and predict words that can replace the current mention of the keyword. If another keyword for this event type appears among the top 50 candidates, we accept this example. The vector representation for this example is the average of the wordpiece tokens that consist the keyword.

The class vector is an average over all the examples for the event type.

2 Solving for M

The following section is a simplified version of the derivation from TapNet (Yoon et al., 2019).

In order to correctly classify c_k, we would like to maximize the dot product with ϕ_k and minimize the dot product with $\phi_{l\neq k}$ in the subspace defined by M. A possible solution would be to find the projection matrix M so that:

$$M(c_k) = \lambda M(\phi_k - \frac{1}{m-1}\sum_{l\neq k}\phi_l) \quad \text{s.t.} \|\phi_i\| = 1, \phi_i^T\phi_{j\neq i} = 0. \tag{1}$$

This implies that

$$M(c_k)^T M(\phi_k) = \lambda\|M\|^2 \quad M(c_k)^T M(\phi_l) = -\lambda\frac{1}{m-1}\|M\|^2 \tag{2}$$

which is a reasonably good separation between the classes.

Let $\hat{\phi}_k = \phi_k - \sum_{l\neq k}\phi_l$, then we can rearrange the previous equation as:

$$M^T(c_k - \lambda\hat{\phi}_k) = 0 \tag{3}$$

Note that this holds for every k. If we define $D \in \mathbf{R}^{d\times n}$ as the matrix with $c_k - \lambda\hat{\phi}_k$ as its kth column, we have $M^T D = \vec{0}$, implying that the columns in M are in the null space of D^T. This null space of D^T can be obtained by QR decomposition.

$$D^T = QR = [Q_1, Q_2]\begin{bmatrix}R_1\\0\end{bmatrix} \tag{4}$$

Although the rank of D is unknown, it will not be larger than n (and with high probability close to n), and thus we can take m columns starting from the $n+1$ column of Q for $M \in \mathbf{R}^{d\times m}$.

In order to account for the new types, we apply some leniency at training time and learn $n' > n$ reference vectors instead of only n vectors for the n classes that appear in the training set. Then when we are asked to identify new types during inference, we update M based on the new class vectors c'_n.

The complete algorithm is listed in Algorithm 1.

2.1 Pseudo Labeling

In the pseudo-labeling process, we compute the token-wise cosine similarity between class vectors and averaged sentence-piece embeddings from

Algorithm 1: Event trigger extraction.

Training;
Input: Label names of n event types. Training examples $\{x, y\}$.
Compute class vectors $c_1, \cdots, c_n$;
Initialize $\phi_1, \cdots, \phi_{n'}$;
for *training episode* **do**
  Compute D;
  Compute M by decomposing D;
  for *training batch* **do**
    Optimize $\theta = \{\Phi, W, \theta_f\}$ w.r.t Equation 8.
  end
end
Testing;
Input: Label names of n' event types.Test examples $\{x\}$.
Result: Set of $\{e, p\}$ event type, position pairs.
Compute class vectors for new classes $c_{n+1}, \cdots, c_{n'}$;
Compute D;
Compute M by decomposing D;
for *test example* **do**
  Predict y for x sequence.
end

BERT-Large. The event type token labels are accepted if the similarity is higher than 0.65 and the O label is assigned if none of the similarity scores are higher than 0.4. For cases in between, we assign an X label which means ignoring the token for loss computation.

3 Dataset Collection and Annotation Details

We removed documents that have less than 100 tokens, and off-topic documents such as excerpts from history books. In the annotation process, annotators can also flag documents as duplicates, or irrelevant. All documents are in English.

When using the KAIROS event ontology, out of the 67 defined event types, we use 51 types that were found in our dataset and merge some rarely seen sub-subevent types. In particular, event sub-subtypes under Contact.Prevarication, Contact.RequestCommand, Contact.ThreatenCoerce were merged. Movement.Transportation.GrantAllowPassage, Transaction.AidBetweenGovernments.Unspecified, Personnel.ChangePosition types were omitted.

Before the event annotation stage, we run a SOTA entity detection model OneIE (Lin et al., 2020) to highlight entity spans. Although this model is not perfect, it can help annotators find candidates for event arguments and reduce annotation time.

The task for the event annotation stage is to identify event trigger and argument spans and label them with the correct event type (argument role). Annotators can also add missing entities or correct the automatic produced entity spans. A two-pass procedure is applied to control the quality of annotation: after annotator A finishes, we randomly assign the annotated document to another more senior annotation B for correction.

After stage 1 finishes, we clean up the annotation by aligning the spans back to word boundaries and then run a joint entity and event coreference system. In stage 2, the annotators are presented with entity (event) clusters and asked to correct them.

Parameter	Value
Base Model	BART-large
Learning rate	[1e-5, 3e-5]
Scheduler	Linear (without warmup)
Batch size	2*8
Max sequence length	512
Training epochs	[3,6]
Beam size	4

Table 1: Hyperparameters for argument extraction

4 Implementation Details

We use the BART-large model (Lewis et al., 2020) for our argument extraction model. Hyperparameters are presented in Table 1. For the zero-shot transfer settings, we trained with a smaller learning rate (1e-5) and more epochs (6). During generation, we use beam search with a beam size of 4. Then we use clarification statements to select the output with the highest probability.

For the trigger extraction task, we used the BERT-large-cased (Devlin et al., 2019) model. The list of hyperparameters as shown in Table 2.

The BERT-CRF model is similar to (Shi and Lin, 2019). To indicate the trigger, we append the trigger to the input sentence: `[CLS] sentence [SEP] trigger [SEP].`

In order to adapt the BERT-QA model for our event ontology, we use the Template 2 (argument based question template) for argument extraction with trigger information: `[wh_word] is the [role name] in [trigger]?`

5 Additional Experiments on ACE

In Tables 4 and 5 we show the complete trigger extraction and argument extraction results on ACE. Entries with an asterisk (*) indicate that these are reported numbers and may be prone to slight differences in dataset splitting and pre-processing.

Parameter	Value
Base Model	BERT-large-cased
Learning rate	3e-5
Weight decay	1e-5
Scheduler	Linear (without warmup)
Batch size	8
Max sequence length	200 (ACE), 400 (WIKIEVENTS)
Training epochs	10
Projection dim	200
Regularization α	0.5

Table 2: Hyperparameters for trigger extraction

Parameter	Value
Base Model	BERT-large-cased
Learning rate	3e-5
Weight decay	1e-5
CRF learning rate	1e-4
Dropout	0.4
Scheduler	Linear (without warmup)
Batch size	8
Max sequence length	200 (ACE), 400 (WIKIEVENTS)
Training epochs	10

Table 3: Hyperparameters for BERT-CRF baseline.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*.

# Seen Event Types	Model	TI Precision	TI Recall	TI F1	TC Precision	TC Recall	TC F1
0	Prototype	1.64	56.03	3.19	1.14	39.01	2.22
	TAPKEY	51.76	59.1	55.19	48.86	55.79	52.10
10 most frequent	Prototype	67.03	72.1	69.48	63.74	68.56	66.06
	BERT-QA	66.25	50.12	57.06	62.5	47.28	53.83
	TAPKEY	67.56	77.78	72.31	64.68	74.47	69.23
1 per general type	Prototype	70.53	66.19	68.29	68.01	63.83	65.85
	BERT-QA	64.91	17.49	27.56	59.64	16.07	25.32
	TAPKEY	66.73	78.72	72.23	63.33	74.7	68.55
All	DYGIE++*	-	-	-	-	-	69.7
	OneIE*	-	-	78.2	-	-	74.7
	BERT-CRF			73.73			69.59
	Prototype	68.1	78.72	73.03	64.83	74.94	69.52
	BERT-QA	68.91	77.54	72.97	65.13	73.29	68.97
	TAPKEY	72.69	76.12	74.36	69.53	72.81	71.13

Table 4: Trigger extraction results on ACE05. Results from DYGIE++ and OneIE are from their papers.

Triggers	Model	AI Precision	AI Recall	AI F1	AC Precision	AC Recall	AC F1
Predicted	DYGIE++*	-	-	55.4	-	-	52.5
	OneIE*	-	-	59.2	-	-	56.8
	BERT-QA*	58.02	50.69	54.11	56.87	49.83	53.12
	BART-Gen	57.57	53.05	55.22	55.99	51.60	53.71
Gold	BERT-QA	69.16	62.65	65.74	66.51	60.47	63.34
	BART-Gen	71.13	68.75	69.92	67.82	65.55	66.67

Table 5: Argument extraction results on ACE05. * indicate reported results.